



TUGAS AKHIR - TF 141581

RANCANG BANGUN PENGENDALIAN pH LARUTAN DENGAN PID *CONTROLLER* PADA BUDIDAYA HIDROPONIK

ZENDY ZITA ANGGANI
NRP. 02311140000085

Dosen Pembimbing
Hendra Cordova, ST, MT.

DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TF 141581

***DESIGN OF SOLVATED PH CONTROLLER WITH
PID CONTROL ON HYDROPONICS SYSTEM***

ZENDY ZITA ANGGANI
NRP. 02311140000085

Supervisor :
Hendra Cordova, ST,MT

DEPARTMENT OF ENGINEERING PHYSICS
Faculty of Industrial Technology
Sepuluh Nopember Institute Of Technology
Surabaya 2018

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini:

Nama : Zendy Zita Anggani
NRP : 02311140000085
Departemen : Teknik Fisika FTI-ITS

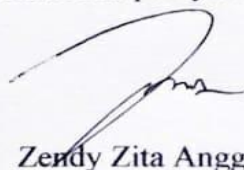
--

Dengan ini menyatakan bahwa tugas akhir saya yang berjudul RANCANG BANGUN PENGENDALIAN pH LARUTAN DENGAN PID *CONTROLLER* PADA BUDIDAYA HIDROPONIK adalah bebas plagiasi. Apabila pernyataan ini terbukti tidak benar, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, Juli 2018

Yang membuat pernyataan,



Zesty Zita Anggani

LEMBAR PENGESAHAN

TUGAS AKHIR

RANCANG BANGUN PENGENDALI pH LARUTAN DENGAN PID CONTROLLER PADA BUDIDAYA HIDROPONIK

Oleh :

Zendy Zita Anggani

NRP : 02311140000085

Surabaya, Juli 2018

Mengetahui / Menyetujui
Dosen Pembimbing



Hendra Cordova, S.T, M.T

NIPN. 196905301994121001



Agus Muhammad Hatta, S.T., M.Si., Ph.D

NIPN. 197809022003121002

LEMBAR PENGESAHAN





RANCANG BANGUN PENGENDALI pH LARUTAN DENGAN PID *CONTROLLER* PADA BUDIDAYA HIDROPONIK

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Rekayasa Instrumentasi
Program Studi-S1 Jurusan Teknik Fisika
Fakultas Teknik Industri
Institut Teknologi Sepuluh Nopember
Oleh :

ZENDY ZITA ANGGANI
NRP. 02311140000085

Disetujui oleh Tim Penguji Tugas Akhir :

1. Hendra Cordova, S.T, M.T..........(Pembimbing)
2. Totok Ruki Biyanto, Ph.D......(Penguji 1)
3. Matradji, M.Sc..........(Penguji 2)
4. Prof. Dr. Ir. Sekartedjo, M.Sc..........(Penguji 3)

SURABAYA, JUNI 2018
Juli, 2018

RANCANG BANGUN PENGENDALI pH LARUTAN DENGAN PID CONTROLLER PADA BUDIDAYA HIDROPONIK

Nama mahasiswa : Zandy Zita Anggani
NRP : 02311140000085
Departemen : Teknik Fisika FTI-ITS
Dosen Pembimbing : Hendra Cordova, S.T., M.T.

Abstrak

Dalam budidaya hidroponik kadar pH air mempengaruhi pertumbuhan tanaman. Hal ini disebabkan karena nutrisi pada hidroponik menggunakan cairan yang nantinya akan diserap oleh akar pada tumbuhan. Penyerapan nutrisi pada tumbuhan dibutuhkan kadar pH yang ditentukan. Untuk menjaga kadar pH maka dilakukannya pengontrolan kadar pH dengan cara menurunkan larutan pH menjadi 6,5. Karena pada kenyataannya pH pada tanaman hidroponik terkadang lebih dari 6,5. Dilakukannya pengujian open loop sebelum memasukkan nilai pada Kp, Ki, dan Kd. Hasil dari pengujian open loop didapatkan nilai PWM maksimal mencapai 29,12 %, *delay time* 16 s, *settling time* 38 s, *overshoot* mencapai 0,12 dan *rise time* 33.

Selanjutnya diberikannya PID pada sistem kontrol yang terdapat tiga macam yaitu dengan memakai Kp, Kp Ki dan Kp Ki Kd. Dari hasil; percobaan yang dilakukan ketika nilai Kp 10 hasil dai respon sitemnya nilai PWM maksimal mencapai 21,12 %, *delay time* 7 s dan *settling time* 11 s. Terjadi *overshoot* mencapai 0,3 dan *rise time* 10 s. Error steady state 0,4. Ketika ditambahkan nilai Ki : 1 hasilnya ialah PWM maksimal mencapai 29,12 %, *delay time* 5 s dan *settling time* 11 s. Terjadi *overshoot* mencapai 0,3 dan *rise time* 9 s. Error steady state 0,1. Ditambahkan nilai Kd 0,5 hasil respon menjadi PWM maksimal 22,63, *delay time* 22 s dan *settling time* 26 s. Error steady state bernilai 0,1. Terjadi *overshoot* mencapai 0,2 dan *rise time* 27 s. Dari hasil respon, hidroponik membutuhkan nilai Kp dan Ki.

Kata kunci : pH(Power of Hidrogen), PID(*Propotional Integral Derivative*), Hidroponik

6

DESIGN OF SOLVATED PH CONTROLLER WITH PID CONTROL ON HYDROPONIC SYSTEM

Name : ZENDY ZITA ANGGANI
NRP : 02311140000085
Departement : *Engineering Physics, FTI-ITS*
Supervisor : Hendra Cordova, S.T, M,T

Abstract

In hydroponic cultivation water pH levels affect plant growth. Because the nutrients in hydroponics use a liquid that will be absorbed by the roots in plants. The absorption of nutrients in plants requires a determined pH level. To maintain the pH level then do the control of pH levels by lowering the pH solution to 6.5. Because in fact the pH in the hydroponic plants is sometimes more than 6.5. Doing open loop testing before entering the values in Kp, Ki, and Kd. The result of open loop test shows that the maximum PWM value reaches 29.12%, delay time 16 s, settling time 38 s, overshoot reach 0.12 and rise time 33. Furthermore, given the PID on the control system there are three kinds that is by using Kp, Kp Ki and Kp Ki Kd. From result; experiments performed when the value of Kp 10 results dai response sitemnya PWM value reaches 21.12% maximum, delay time 7 s and settling time 11 s. An overshoot occurs at 0.3 and a 10 s rise time. Steady state error 0.4. When added value Ki: 1 result is maximum PWM reach 29,12%, delay time 5 s and settling time 11 s. An overshoot reaches 0.3 and a 9 s rise time. Steady state error 0.1. Added value of Kd 0,5 result response to PWM maximum 22,63, delay time 22 s and settling time 26 s. Steady state error is 0.1. An overshoot reaches 0.2 and a 27 s rise time. From the response result, hydroponics need the value of Kp and Ki.

***Key word : pH(Power of Hidrogen), PID(Propotional Integral
Derivative), hydroponic***

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas berkat, rahmat dan kebesaran-Nya sehingga saya selaku penulis dapat menyelesaikan Tugas Akhir sampai dengan penyusunan Laporan Tugas Akhir yang berjudul **“Rancang Bangun Pengendalian pH larutan dengan PID Controller pada budidaya hidroponik ”**.

Tugas akhir ini merupakan persyaratan akademik yang harus dipenuhi dalam Program Studi S-1 Teknik Fisika FTI-ITS. Untuk itu penulis mengucapkan terima kasih kepada :

1. Allah SWT yang telah memberikan segala hidayahnya sehingga tugas akhir ini dapat terselesaikan dengan baik
2. Keluarga tercinta, Bapak, Mama, dan adik-adik atas kasih sayang, kesabaran, dan menjadi motivasi terbesar dalam menyelesaikan studi dan tugas akhir di Teknik Fisika ini.
3. Bapak Agus Muhamad Hatta ST,Msi, Ph.D selaku Ketua Departemen Teknik Fisika.
4. Bapak Hendra Cardova ST,MT selaku dosen pembimbing tugas akhir.
5. Ibu Prof. DR. Aulia Siti Aisjah, Ir. MT. yang telah menjadi dosen wali selama perkuliahan di Teknik Fisika.
6. Terima kasih untuk teman – teman TF angkatan 2011 dan angkatan lain selalu memberi semangat, bantuan dan dukungan selama ini
7. Dan untuk semua orang yang telah membantu sehingga tugas akhir ini dapat terselesaikan.

Penulis menyadari bahwa karya yang sempurna hanya ada pada Allah SWT. Oleh sebab itu, penulis sangat berterimakasih atas segala masukan, kritik dan saran yang membangun dari pembaca agar laporan ini menjadi lebih baik dari sebelumnya.

Demikian laporan ini penulis buat, semoga laporan ini dapat memberikan manfaat selain bagi penulis sendiri, dan bagi pembaca sekalian.

Surabaya, Juli 2018

Penulis

DAFTAR ISI

| | |
|---------------------------------------|------|
| HALAMAN JUDUL | I |
| LEMBAR PENGESAHAN | iii |
| ABSTRAK | v |
| ABSTRACT | vi |
| KATA PENGANTAR | vii |
| DAFTAR ISI | ix |
| DAFTAR GAMBAR | xi |
| DAFTAR TABEL | xiii |
| Daftar Gambar | xii |
| Daftar Tabel | xiii |
| BAB I | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Rumusan Masalah | 1 |
| 1.3. Tujuan..... | 2 |
| 1.4. Lingkup Kerja | 2 |
| 1.5. Manfaat..... | 2 |
| BAB II | 3 |
| 2.1. Budidaya Hidroponik | 3 |
| 2.2. Teori pH..... | 4 |
| 2.2.1. Pengukuran pH | 4 |
| 2.2.2. Sistem Pengendali pH | 5 |
| 2.3. Komponen Sistem Pengendali | 6 |
| 2.3.1. Mikrokontroler..... | 6 |
| 2.3.2. Sensor pH | 7 |
| 2.3.3. Driver Motor..... | 8 |

| | |
|---|----|
| 2.3.4. Catu Daya | 8 |
| 2.4. PID (<i>Propotional integral derivative</i>) | 8 |
| BAB III | 11 |
| 3.1. Alur Penelitian | 11 |
| 3.2. Studi Literatur | 12 |
| 3.3. Perancangan Hidroponik | 12 |
| 3.4. Perancangan PID | 14 |
| BAB IV | 15 |
| 4.1. Uji Karakteristik Alat | 15 |
| 4.1.1 Kalibrasi Sensor | 15 |
| 4.2. Hasil pengujian <i>open loop</i> | 17 |
| 4.3. Hasil pengujian <i>close loop</i> | 17 |
| BAB V | 21 |
| 5.1. Kesimpulan | 21 |
| 5.2. Saran | 21 |
| DAFTAR PUSTAKA | 23 |
| LAMPIRAN A | 24 |
| LAMPIRAN B | 26 |
| LAMPIRAN C | 34 |

Daftar Gambar

| | |
|--|----|
| Gambar 2.1 Arduino Mega | 6 |
| Gambar 2.2 Catu Daya..... | 8 |
| Gambar 2.3 Blok Diagram Kontrol PID | 9 |
| Gambar 3.1 Diagram Alir Tahapan Penelitian | 11 |
| Gambar 4.1 Pengujian pH naik | 16 |
| Gambar 4.2 Pengujian pH turun | 16 |
| Gambar 4.3 Grafik pengujian <i>open loop</i> | 17 |
| Gambar 4.4 Respon kadar pH menuju 6,5 Kp 10 | 18 |
| Gambar 4.5 Respon kadar pH menuju 6,5 Kp 10 Ki 1 | 18 |
| Gambar 4.6 Grafik PID nilai kadar pH menuju 6,5 Kp 10 | 19 |

Daftar Tabel

| | |
|--|----|
| Tabel 2.1 Spesifikasi Arduino Mega | 9 |
| Tabel 4.1 Hasil Kalibrasi Sensor pH | 17 |

BAB I

PENDAHULUAN

1.1. Latar Belakang

Budidaya hidroponik menjadi salah satu pilihan saat ini terutama di daerah perkotaan. Hidroponik memiliki beberapa keuntungan, antara lain terdapat kemudahan dalam pengelolaan, efisiensi jumlah nutrisi atau pupuk, jumlah air, dan juga dapat dikembangkan secara komputerisasi [6]. Teknologi tanam hidroponik adalah suatu sistem bercocok tanam tanpa menggunakan air sebagai media tanamnya. Air yang digunakan hendaknya memenuhi syarat- syarat tertentu , salah satunya ialah pH [6].

Kadar pH yang terdapat pada aliran air dimana kadar pH tersebut akan berubah seiring penambahan nutrisi. Kadar pH yang tidak stabil dapat menyebabkan pertumbuhan tanaman menjadi lambat dan tidak subur. Kadar pH harus sesuai dengan yang di harapkan [10].

Pada saat ini pengembangan mengenai kontrol pH hidroponik sudah mulai dikembangkan, tetapi rata - rata di Indonesia masih menggunakan kontrol manual, yaitu melihat nilai pH hidroponik menggunakan pH meter digital. Untuk itu penulis bermaksud merancang alat kontrol pH hidroponik berbasis arduino dengan metode PID.

Pada alat kontrol pH hidroponik, dapat mensetting nilai pH yang diinginkan sesuai dengan karakter tiap tanaman. Alat kontrol ini menggunakan arduino mega sebagai yang mengatur alat kontrol ini. Metode kontrol yang digunakan ialah PID yaitu *propotional*, *integral* dan *Derivative*.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan, dapat dirumuskan masalah dalam penelitian tugas akhir ini adalah:

- 1) Bagaimana merancang sistem kontrol pH pada sistem hidroponik dengan mengimplementasikan metode kontrol PID?
- 2) Bagaimana pengaruh perubahan nilai K_p , K_i dan K_d terhadap nilai pH yang diinginkan?

1.3. Tujuan

Dari rumusan masalah yang didapat, maka tujuan dari penelitian tugas akhir ini adalah:

- 1) Merancang sistem kontrol pH pada sistem hidroponik dengan mengimplementasikan metode PID.
- 2) Mengetahui pengaruh perubahan nilai K_p , K_i dan K_d terhadap nilai pH yang diinginkan

1.4. Lingkup Kerja

Adapun lingkup penelitian tugas akhir yang dikerjakan meliputi:

- 1) Pengontrolan kadar pH tidak dilakukan hingga masa panen tanaman.
- 2) Pembahasan tidak termasuk dengan penambahan nutrisi tanaman..

1.5. Manfaat

Manfaat yang didapatkan dari penelitian tugas akhir ini adalah:

- 1) Membantu dan mempermudah kerja petani hidroponik dalam melakukan *monitoring* dan kontrol pH pada air hidroponik.
- 2) Meningkatkan hasil panen dari budidaya hidroponik.

BAB II

TINJAUAN PUSTAKA

2.1. Budidaya Hidroponik

Budidaya hidroponik pada saat ini sering diminati beberapa masyarakat. Karena hidroponik ini merupakan salah satu metode menanam tanaman tanpa menggunakan media air sebagai media utamanya. Penanaman dilakukan dengan menggunakan wadah yang dialiri oleh air. Nantinya tanaman akan disangga dengan bahan berupa kerikil, batu- batu kecil, rockwool dan masih banyak lagi jenis bahan yang bisa dimanfaatkan.

Pada budidaya hidroponik dikarenakan media air sebagai media utama sehingga terdapat beberapa hal penting yang harus diketahui. Faktor utama yang mempengaruhi perkembangan tanaman dalam budidaya ini adalah tersedia unsur nutrisi penunjang yang sesuai dengan jenis umur tanaman dan kadar pH yang terkandung pada aliran air [6].

pH sendiri adalah ukuran kadar keasaman atau basa suatu larutan dengan memperhitungkan konsentrasi ion hidrogen dalam larutan tersebut. Kadar pH air mempengaruhi pertumbuhan pada tanaman. Karena kadar pH tersebut akan mempengaruhi akar dalam menyerap nutrisi yang terdapat di dalam air. Nantinya ketika kadar pH yang terdapat pada aliran air tidak diatur sesuai yang diinginkan menyebabkan terjadinya keterlambatan pertumbuhan pada tanaman. Beberapa pengaruh pH pada hidroponik yaitu:

1. kadar pH mempengaruhi daya serap tanaman.
2. Jika nilai pH berubah ke arah yang asam sekitar 3.5- 4.5 maka tanaman cenderung mudah terkena penyakit
3. Jika pH berubah ke arah basa diatas 7 maka dapat berpengaruh terhadap berkurangnya ketersediaan zat besi (Fe), Mangan (Mn), tembaga (Cu), Zinc (Zn) dan boron (B)
4. Jika pH dibawah 6.0 menyebabkan turunnya daya larut asam fosfat, kalsium dan magnesium

Terdapat larutan yang dapat digunakan untuk menurunkan kadar pH yaitu menggunakan pH down atau dapat menggunakan larutan asam

seperti asam sulfat maupun asam asetat. Sedangkan larutan yang digunakan untuk menaikkan kadar pH yaitu larutan pH up atau dapat menggunakan 1 botol soda kue yang dilarutkan kedalam 100 ml air biasa. Soda kue merupakan larutan natrium bikarbonat yang larut dalam air dan dapat menurunkan kadar asam.

2.2. Teori pH

pH (*power of hydrogen*) atau derajat keasaman ialah tingkat keasaman atau basa yang dimiliki oleh suatu zat, larutan maupun benda. pH normal ialah pH yang bernilai 7 sementara pH dikatakan basa jika $\text{pH} > 7$. Sedangkan $\text{pH} < 7$ dinyatakan bersifat asam. pH yang memiliki keasaman tinggi yaitu pH 0 sedangkan yang memiliki kebasaan tinggi ialah pH 14. Menentukan nilai pH yang dilakukan dengan cara manual biasanya memakai kertas lakmus. Yang nantinya ketika kertas lakmus diberikan cairan yang ingin diketahui kadar pH akan berubah warna. Menjadi warna merah jika keasamannya tinggi sedangkan menjadi warna biru jika keasamannya rendah.

Asam dan basa juga dapat dilihat nilainya dengan menggunakan pH meter. pH meter ini sendiri bekerja berdasarkan prinsip elektrolit atau konduktifitas suatu larutan. Sistem pengukurannya terbagi menjadi tiga bagian yaitu elektroda pengukuran pH, elektroda referensi dan alat pengukur inpendasi tinggi. pH sendiri adalah negative logaritma dari aktivitas ion hidrogen.

$$\text{pH} = -\log[\text{H}^+] \quad (2.1)$$

asam dan basa adalah besaran yang sering digunakan untuk pengolahan suatu zat. Pada bidang pertanian keasaman pada air ataupun tanah perlu diketahui.

2.2.1. Pengukuran pH

Pada prinsipnya pengukuran pH didasarkan pada potensoal elektro kimia yang terjadi antara larutan yang terdapat di dalam

elektroda gelas yang telah diketahui dengan larutan yang terdapat diluar elektroda gelas. Hal ini karena lapisan tipis dari gelembung kaca akan berinteraksi dengan ion hidrogen yang ukurannya relatif kecil dan aktif, elektroda tersebut akan mengukur potensial elektrokimia dari ion hidrogen atau diistilahkan dengan *potensial of hydrogen* [8].

Sensor pH akan mengukur potensial listrik antara merkuri Chloride (HgCl) pada elektroda pembanding dan potassium chloride (KCl) yang merupakan larutan didalam gelas electrode serta potensial antara larutan dan elektroda perak. Tetapi potensial antara sampel yang tidak diketahui dengan elektroda gelas dapat berubah tergantung sampelnya, oleh karena itu perlu dilakukan kalibrasi dengan menggunakan larutan yang ekuivalen yang lainnya untuk menetapkan nilai dari pH. Elektroda pembanding calomel terdiri dari tabung gelas yang berisi potassium kloride (KCl) yang merupakan elektrolit yang mana terjadi kontak dengan merkuri chloride (HgCl) diujung larutan KCl. Tabung gelas ini mudah pecah sehingga untuk menghubungkannya digunakan ceramic berpori atau bahan sejenisnya. Elektroda semacam ini tidak mudah terkontaminasi oleh logam dan unsure natrium.

Elektroda gelas terdiri dari tabung kaca yang kokoh yang tersambung dengan gelembung kaca tipis yang didalamnya terdapat larutan KCl sebagai buffer pH 7. Elektroda perak yang ujungnya merupakan perak kloride (AgCl₂) dihubungkan kedalam larutan tersebut. Untuk meminimalisir pengaruh electric yang tidak diinginkan, alat tersebut dilindungi oleh suatu lapisan kertas pelindung yang biasanya terdapat dibagian dalam elektroda gelas. Salah satu contoh bentuk elektroda gelas dari jenis sensor pH dapat dilihat pada.

2.2.2. Sistem Pengendali pH

Agar dapat menjaga pH air pada hidroponik dibutuhkan sebuah sistem kontrol yang handal. Sistem kontrol *closed loopfeedback* digunakan untuk bisa melakukan eliminasi kesalahan pada plan dan mengatasi adanya *disturbance* atau gangguan [9]. Caranya ialah dengan terus memberikan nilai terukur dari variable yang ingin

dikendalikan kepada controller. Jadi apabila ada kesalahan akan terus dikoreksi.

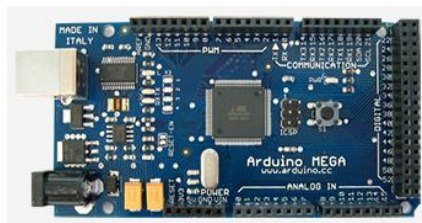
Kontrol pH mulai melakukan pengukuran dari variable proses(PV) yang ingin dikendalikan yaitu pH terlarut menggunakan sensor. Nilai pengukuran akan dibandingkan dengan setpoint. Nantinya didapatkan nilai error. Hasil dari nilai error tersebut akan masuk ke controller sebagai input dan kemudian menjadikan sebuah perintah yang berupa variable manipulasi(MV) ke aktuator.

2.3. Komponen Sistem Pengendali

Dibutuhkan komponen sistem pengendalian berupa, mikrokontroler, sensor, *driver* motor, serta komponen pendukung seperti catu daya.

2.3.1. Mikrokontroler

Mikrokontroler merupakan *single chip* yang terdapat pada komputer atau berada pada divais kecil mikrokontroler yang digunakan untuk melakukan aksi pada sebuah objek, proses atau perintah. Mikrokontroler disebut sebagai *single chip* dikarenakan pada mikrokontroler mempunyai memori dan perangkat I/O yang berada di dalamnya [2]. Mikrokontroler yang digunakan adalah arduino Mega.



Gambar 2.1 Arduino Mega [1]

Arduino mega adalah sebuah mikrokontroler berbasis Atmega2560. Arduino mega memiliki pin *input/output* sebanyak 54 (14 diantaranya bisa digunakan sebagai *output* PWM), pin input analog sebanyak 16, pin UARTs 4, sebuah *crystal oscillator* 16 MHz,

koneksi USB, *power jack*, *ICSP header*, dan tombol reset. Untuk menghidupkan sistem, arduino mega hanya memerlukan catu daya yang bisa diambil dari USB komputer atau adapto AC/DC atau menggunakan baterai.

Tabel 2.1Spesifikasi Arduino Mega

| | |
|------------------------------------|--------------------------------------|
| <i>Microcontroller</i> | Atmega2560 |
| <i>Operating Voltage</i> | 5V |
| <i>Input Voltage (recommended)</i> | 7-12V |
| <i>Input Voltage (limits)</i> | 6-20V |
| <i>Digital I/O Pins</i> | 54 (15 bisa untuk keluaran PWM) |
| <i>Analog Input Pins</i> | 16 |
| <i>DC Current per I/O Pin</i> | 20 mA |
| <i>DC Current for 3.3V Pin</i> | 50 mA |
| <i>Flash Memory</i> | 256 KB, 8 KB untuk <i>bootloader</i> |
| <i>SRAM</i> | 8 KB |
| <i>EEPROM</i> | 4 KB |
| <i>Clock Speed</i> | 16 MHz |

2.3.2. Sensor pH

Sensor Ph yang digunakan ialah sensor ph SEN 0161. Sensor ini dilengkapi dengan module yang berfungsi untuk mengirimkan data analog ke arduino dengan prinsip kerja membandingkan larutan yang terdapat dalam probe yaitu larutan HCl dengan larutan didalam wadah yang tidak diketahui hingga module menghasilkan tegangan yang berbeda. Nilai tegangan akan dikirim ke arduino yang nantinya akan di olah dan mengeluarkan nilai kadar pH. Bahan yang terdapat pada probe sensor ph yaitu ujung elektroda kaca yang berbentuk bulb. Pada Bulb ini dipasangkan dengan silinder kaca non- konduktor atau plastik memanjang.

2.3.3. Driver Motor

Driver motor ini berfungsi untuk mengatur kecepatan pada pompa motor. Didalam rangkaian driver motor ini terdapat ic l293d dengan 4 buah pin digital input, 2 pin pwm (0-255) dan 4 pin sebagai output yang mampu mengendalikan 2 motor dc. Driver motor ini akan bekerja dengan kapasitas arus 2A dengan tegangan 12 volt untuk dua motor dc. Pada aksi modul driver, terdapat dua yaitu low atau high pada input digital dan memberikan signal pwm (0-255) enable 1 atau 2 untuk mengatur kecepatan motor dc.

2.3.4. Catu Daya

Adaptor adalah sebuah rangkaian yang berfungsi untuk mengubah tegangan AC menjadi tegangan DC. Adaptor yang digunakan adalah jenis adaptor *power supply* yang mengubah tegangan 220 volt AC menjadi tegangan 5 volt DC, sesuai dengan kebutuhan sensor, rangkaian pengkondisian sinyal, relay, dan mikrokontroler. Sedangkan untuk pompa aerator menggunakan tegangan AC yang dikontrol melalui relay.



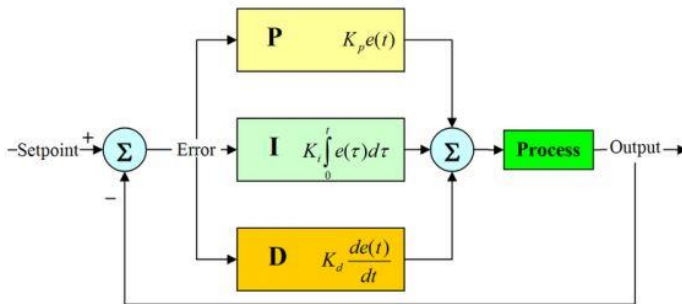
Gambar 2.2 Catu Daya

2.4. PID (*Proportional integral derivative*)

Kontrol *Proportional, Integral, Derivative* (PID) adalah sistem pengendali yang umum digunakan di industri Gambar 1 menunjukkan blok diagram kontrol PID. Kontrol *Proportional* (P), *Integral* (I), dan *Derivatif* (D) dapat digunakan bersamaan secara paralel ataupun digunakan terpisah dengan tidak menggunakan salah satu komponen P, I atau D.

$$u(t) = K_P e(t) + \frac{K_P}{T_i} \int_0^t e(t) dt + K_P T_d \frac{de(t)}{dt} \quad (2.2)$$

Persamaan pada 2.2 merupakan bentuk dari sistem pengendalian PID. Parameter pada setiap elemen pengendali saling mempengaruhi satu dengan yang lainnya. Secara diagram blok dapat juga dilihat pada gambar 2.3



Gambar 2.3 Blok Diagram Kontrol PID

Kontrol *proporsional* memiliki *output* yang sebanding atau *proporsional* dengan besar sinyal kesalahan (selisih antara nilai yang diinginkan dengan nilai aktualnya, error). Setiap saat terjadinya perubahan pada sinyal input menyebabkan sistem akan langsung mengeluarkan sinyal keluar sebesar konstanta pengalinya [8].

Sedangkan pada pengontrol *integral* berfungsi untuk menghilangkan error *steady-state* menjadi nol. Jika sebuah *plant* tidak mempunyai unsur *integrator* (1/s), pengontrol *proporsional* tidak mampu menjamin output sistem akan tepat sesuai respon yang diinginkan, sehingga dibutuhkan pengontrol *integral* [8].

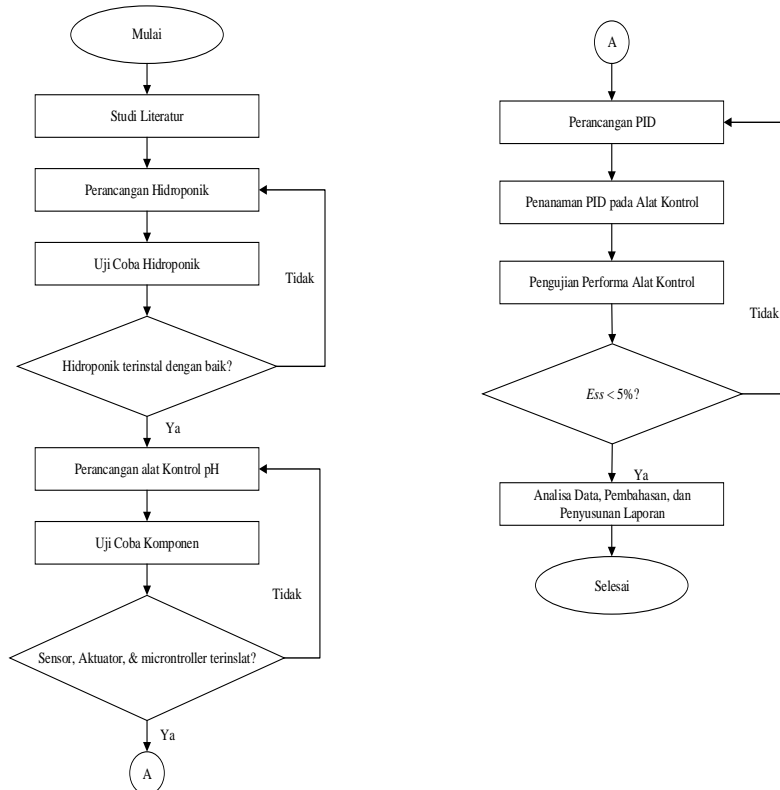
Besar output dari pengontrol *derivatif* memiliki sifat seperti operasi diferensial pada umumnya. Pengontrol *derivatif* menggunakan kecepatan perubahan sinyal kesalahan sebagai parameter pengontrol. Apabila tidak ada perubahan sinyal error, maka *output* dari kontrol derivatif tidak akan berubah [8].

Halaman ini memang dikosongkan

BAB III METODOLOGI PENELITIAN

3.1. Alur Penelitian

Secara garis besar, alur dari penelitian yang dilakukan tercantum pada diagram alir berikut:



Gambar 3.1Diagram Alir Tahapan Penelitian

Pertama kali hal yang dilakukan pada penelitian ini ialah melakukan studi literatur. Studi literatur dilakukan dengan cara mencari dasar permasalahan dan dasar pemecahan masalah

berdasarkan jurnal dan penelitian sebelumnya. Ketika studi literatur selesai dilanjutkan dengan perancangan hidroponik yang nantinya akan di uji coba dalam penelitian kali ini. Ketika selesai perancangan hidroponik dilakukan pengecekan kembali untuk memastikan hidroponik berjalan dengan baik. Dilanjutkan dengan perancangan alat kontrol pH yang mencakupi mekanik, elektrik, dan program). Setelahnya dilakukan pengujian alat kontrol pH. Pengujian dilakukan dengan cara mengkalibrasikan sensor pH dan mengontrol nilai pH dalam larutan menggunakan penambahan larutan pH down.

Perancangan PID yang nantinya akan dimasukkan kedalam alat kontrol. Melakukan pengujian PID alat kontrol yang nantinya memastikan apabila masih kesalahan (ess) masih di atas batas yang ditentukan yaitu 5% perbaikan algoritma kontrol harus dilakukan, dan di uji kembali sampai ess sesuai yang diinginkan.

3.2. Studi Literatur

Berdasarkan hasil studi literatur, *setpoint* kadar pH yang diinginkan yang baik sekitar 7.0. untuk pengujian controllertoleransi error yang diijinkan adalah 5 % dari setpoint. Melakukan beberapa pengujian PID sampai mendapatkan nilai k_p , k_i , dan k_d yang tepat untuk hidroponik.

3.3. Perancangan Hidroponik

Hidroponik dirancang dengan ukuran kecil dikarenakan untuk mempermudah uji coba alat dan pengamatan. Pada bagian hidroponik ini terdapat tempat menanam dan wadah penyimpanan air yang berisi nutrisi yang akan dikontrol nilai pH -nya. Ukuran tempat penanaman adalah 50 x 30 x 14 cm. Sedangkan ukuran penyimpanan air 60 x 30 x 30 cm. Wadah penampungan air akan diisi dengan 30 liter air yang nantinya akan dipompa ke tempat penanaman. Sedangkan dalam tempat penanaman menggunakan media tanam pasir malang yang diisinya sampai dengan ketinggian 12 cm.

Sistem yang digunakan pada hidroponik ialah *ebb and flow*. Sistem ini berjalan dengan cara pompa akan menaikkan air dari wadah penyimpanan sampai ke tempat tanam. Setelah air terisi sampai dengan batas ketinggian yang diinginkan akan terjadi pembuangan air kebawah menuju wadah penampungan. Air akan mengalir kewadah penampungan sampai dengan batas yang diinginkan. Pada hidroponik ini batas air yang diisi pada wadah tanaman sampai pada ketinggian 12.cm dan batas bawahnya sampai dengan 3 cm.

Terdapat penambahan pompa pada wadah penampungan. Hal ini dilakukan untuk mempermudah tercampurnya air dengan pH down nantinya ketika kontrol pH sudah dijalankan. Kedua pompa akan dinyalakan setiap 45 menit dalam waktu satu jam.

3.4 Perancangan Alat Kontrol pH

Perancangan alat kontrol pH bekerja dengan mengukur kadar pH dalam kolam menggunakan sensor pH. Terdapat variable temperatur yang diukur. Temperature diukur dikarenakan akan ada pengaruh nilai dari pembacaan . Dalam pengkalibrasian sensor juga mengukur temperatur pada kolam. Yang nantinya hasil pembacaan akan diolah.

Sensor pH sendiri hasilnya akan di baca oleh mikrokontroler. Pada mikrokontroller sendiri sudah dimasukkan *coding* untuk melakukan aksi mengontrol kadar pH. Jika *setpoint* yang diinginkan tidak sesuai dengan keadaan air maka mikrokontroler akan mengirimkan sinyal PWM pompa motor. Pengiriman ini melalui *driver* motor yang nantinya memberikan daya kepada pompa motor. Pada *driver* motor sendiri diberikan tegangan 12 v DC. Hal ini dilakukan agar motor dapat bekerja secara maksimal.

Data pembacaan pH akan ditampilkan pada LCD. Selain itu hasil pembacaan akan disimpan pada *sd card*. Ditambahkannya *keypad* pada alat kontrol yang digunakan untuk mengubah nilai pH ataupun nilai K_p , K_i , dan K_d pada kontrol PID.

3.4. Perancangan PID

Ketika alat kontrol sudah siap, diberikan algoritma PID yang nantinya akan memberikan respon terhadap nilai pH pada wadah air. Pada perancangan PID ini menentukan besaran nilai K_p , K_i dan K_d . Percobaan yang dilakukan pertama kali menggunakan nilai K_p . Selanjutnya ditambahkannya nilai K_i . Dan yang terakhir menggunakan nilai K_p , K_i dan K_d .

Hasil dari pembacaan akan mendapatkan nilai MV (Manipulated Variable), *delay time*, *rise time*, *overshoot*, *settling time* dan *error steady state*. Hasil dari semua percobaan akan dibandingkan. Didapatkan nilai pada K_p , K_i dan K_d yang baik digunakan pada hidroponik.

BAB IV

ANALISA DATA DAN PEMBAHASAN

4.1. Uji Karakteristik Alat

Pengujian karakteristik dilakukan dengan mengkalibrasi sensor pH. Pengujian dilakukan untuk mengetahui keakuratan dari pembacaan sensor pH yang digunakan.

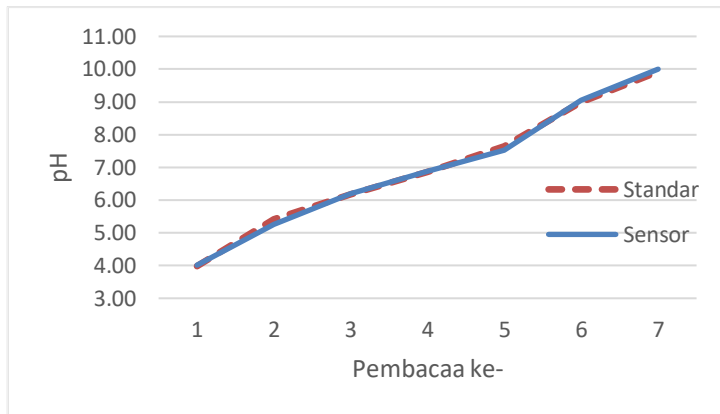
4.1.1 Kalibrasi Sensor

Pengujian dilakukan dengan cara membandingkan hasil pembacaan sensor pH dengan larutan pH standar . Pengujian ini dilakukan di Laboratorium Instrumentasi dan kontrol Teknik Fisika FTI ITS.

Tabel 4.1 Hasil Kalibrasi Sensor pH

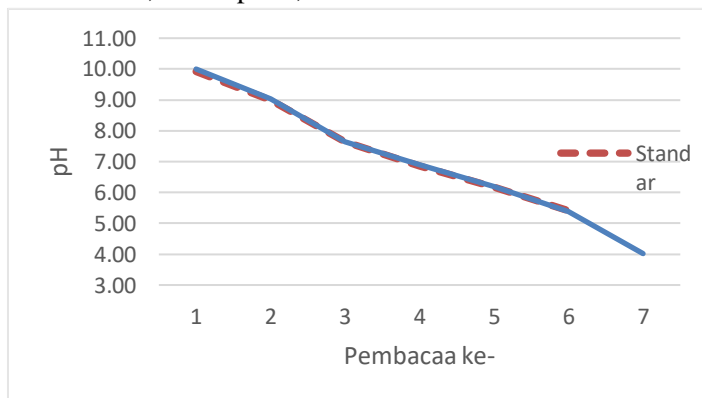
| Standar (pH) | I | II | III | IV | V | Rata-rata |
|-----------------|------|------|------|------|------|-----------|
| 4 | 4,00 | 3,98 | 3,98 | 3,97 | 3,98 | 3,98 |
| 5 | 5,40 | 5,41 | 5,41 | 5,41 | 5,41 | 5,41 |
| 6 | 6,17 | 6,18 | 6,17 | 6,18 | 6,19 | 6,18 |
| 7 | 6,85 | 6,85 | 6,85 | 6,85 | 6,85 | 6,85 |
| 8 | 7,65 | 7,64 | 7,65 | 7,65 | 7,65 | 7,65 |
| 9 | 9,00 | 9,00 | 9,00 | 9,00 | 9,00 | 9,00 |
| 10 | 9,92 | 9,92 | 9,92 | 9,92 | 9,92 | 9,92 |

Hasil dari kalibrasi dapat dilihat pada tabel 4.1. Terlihat bahwa Hasil dari pembacaan sensor pH dengan larutan pH standar sangat mendekati. Hanya pada larutan ph 5 mencapai eror 0,41. Dari data yang didapatkan rata- rata eror 0,1. Selanjutnya dilakukan pengujian dengan cara membandingkan pembacaan nilai pH dengan cara pengambilan nilai pH dari pH 4 sampai pH 7 mulai dari nilai pH yang terkecil sampai yang terbesar, begitu pula sebaliknya. Hasil data pengujian dapat di lihat pada gambar 4.1 untuk pengujian pH naik dan pada gambar 4.2 untuk pengujian pH turun.



Gambar 4.1 Pengujian Ph naik

Pengujian ini dilakukan 5 kali pengambilan data pada setiap nilai pH. Yang dari hasil kelima data tersebut diambil rata-rata pembacaan. Dari hasil pengujian ph 4 sampai ke pH 10 didapatkan rata-rata eror pada sensor pH berkisar 0,02 sampai dengan 0,14. Sedangkan pada pengujian ph turun didapatkan rata-rata eror berkisar antara 0,02 sampai 0,08.



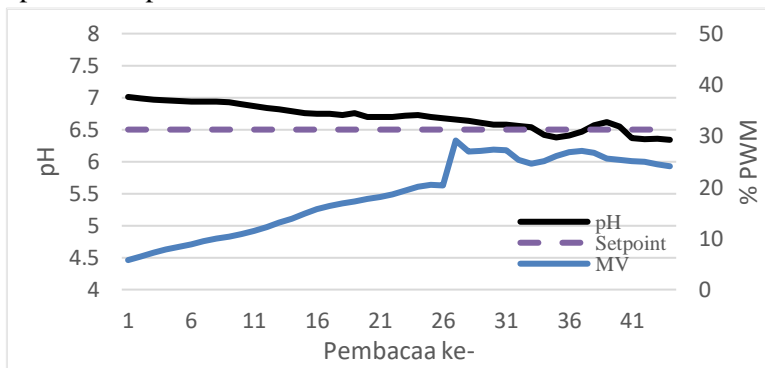
Gambar 4.1 Pengujian ph turun

Dari nilai hasil kalibrasi yang didapatkan alat sensor pH memiliki eror yang masih dalam batas wajar. Terjadinya penurunan

ataupun kenaikan pH yang dibaca tidak membuat error pembacaan bertambah.

4.2. Hasil pengujian *open loop*

Pengujian open loop dilakukan dengan cara menurunkan pH 7 mencapai nilai pH 6,5 tanpa menggunakan kontrol PID. Hasil dari pembacaan kontrol *open loop* ini akan mendapatkan karakter dari alat pengendali pH terhadap *plant* hidroponik. Hasil dari pembacaan dapat dilihat pada Gambar 4.3.

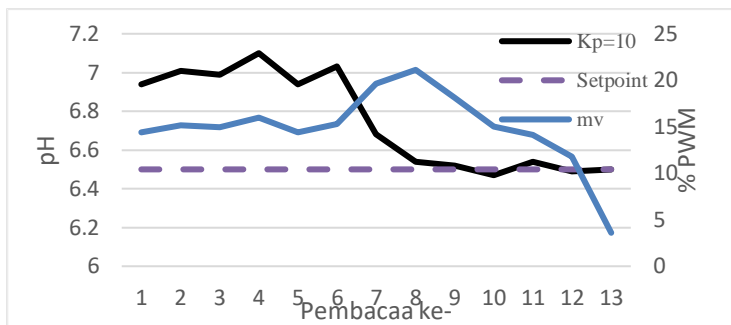


Gambar 4.3 Grafik pengujian *open loop*

Dengan ditambahkannya larutan ph down untuk menurunkan nilai pH dengan cara menyalakan pompa motor. Keluaran pompa motor dilihat dengan nilai PWM yang diberikan dengan presentasi 0 sampai 100 %. Nilai PWM maksimal mencaai 29,12 %. Pada pengujian *open loop* didapatkan delay time 16 s dan *settling time* 38 s. Terjadi overshoot mencapai 0,12 dan rise time 33 s. Γ pengontrolan kadar pH air dengan metode PID

4.3. Hasil pengujian *close loop*

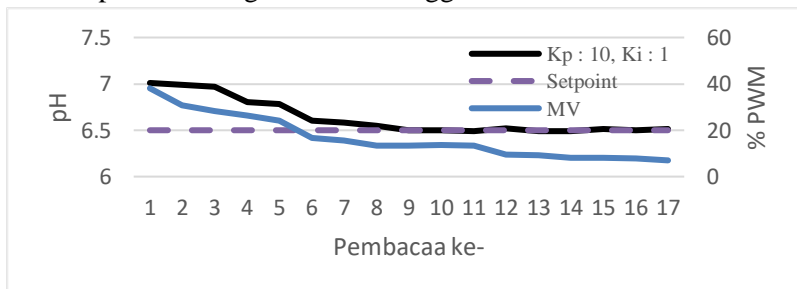
Pengujian kadar pH terhadap setpoint yang diinginkan dari perhitungan PID. Pengujian pertama dilakukan dengan memberikan nilai Kp 10 dengan *setpoint* yang diinginkan adalah 6.5. kondisi ph awal adalah 7. Hasil pembacaan dapat dilihat pada Gambar 4.4.



Gambar 4.4 Respon pH 7 menuju 6,5 dengan kp 10

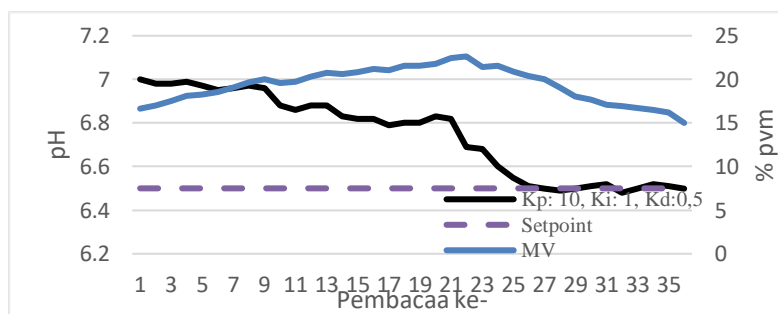
Nilai PWM maksimal mencapai 21,12 %. Nilai ini lebih kecil dibandingkan dengan *open loop*. Sedangkan *delay time* 7 s dan *settling time* 11 s. *Delay time* dan *settling time* lebih cepat dibandingkan dengan *open loop*. Terjadi *overshoot* mencapai 0,3 dan *rise time* 10 s. Hasil Error steady state pada pemberian nilai Kp ini ialah 0,4. Dari penambahan nilai pada Kp terlihat berkurangnya *rise time*. Ketika selisih antara *rise time open loop* terhadap penambahan nilai Kp ini mencapai 23 s.

Pengujian kedua dengan cara ditambahkan nilai Ki pada kontrol PID. Hasil dari kontrol PID dengan nilai Kp 10 dan Ki 1 dapat dilihat pada gambar 4.5. Dari grafik terlihat nilai PWM maksimal mencapai 29,12 %. Nilai PWM ini lebih besar dibandingkan dengan kontrol tanpa menggunakan Ki. Pada *delay time* 5 s dan *settling time* 11 s. Untuk *delay time* dan *settling time* lebih cepat dibandingkan tidak menggunakan Ki.



Gambar 4.5 Respon pH 7 menuju 6,5 dengan Kp 10, dan Ki 1

Nilai *overshoot* pada pengendalian PID menggunakan nilai K_p dan K_i ialah 0,3 dan *rise time* yang terjadi ialah 9 s. Untuk nilai pada *overshoot* sama dengan pengujian sebelumnya tetapi pada *settling time* lebih cepat. Nilai Error steady statenya ialah 0,1. Nilai tersebut lebih kecil dibandingkan percobaan sebelumnya.



Gambar 4.6 Respon pH 7 menuju 6,5 dengan k_p 10, k_i 1 dan k_d 0,5

Pengujian selanjutnya dengan menambahkan nilai K_d pada kontrol PID. Sehingga PID yang digunakan dengan nilai K_p 10, K_i 1 dan K_d 0,5. Dapat dilihat hasil dari pengujian tersebut pada Gambar 4.6. Didapatkan nilai PWM maksimal 22,63. Peningkatan nilai PWM perlahan. Untuk *delay time* 22 s dan *settling time* 26 s. *Error steady state*nya ialah 0,1. Terjadi *overshoot* mencapai 0,2 dan *rise time* 27 s

Halaman ini memang dikosongkan

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil dari analisa data yang dilakukan, didapatkan kesimpulan dari tugas akhir ini adalah sebagai berikut:

- 1) Pengujian *open loop* didapatkan nilai PWM maksimal mencapai 29,12 %, *delay time* 16 s, *settling time* 38 s. Terjadi *overshoot* mencapai 0,12 dan *rise time* 33 s.
- 2) Menggunakan nilai k_p 10 mendapatkan hasil nilai PWM maksimal mencapai 21,12 %, *delay time* 7 s dan *settling time* 11 s. Terjadi *overshoot* mencapai 0,3 dan *rise time* 10 s. Error steady state 0,4.
- 3) Menggunakan nilai K_p 10 dan K_i 1 didapatkan hasil nilai PWM maksimal mencapai 29,12 %, *delay time* 5 s dan *settling time* 11 s. Terjadi *overshoot* mencapai 0,3 dan *rise time* 9 s. Error steady state 0,1.
- 4) Menggunakan nilai K_p 10, K_i 1 dan K_d 0,5 didapatkan nilai PWM maksimal 22,63, *delay time* 22 s dan *settling time* 26 s. Error steady statenya ialah 0,1. Terjadi *overshoot* mencapai 0,2 dan *rise time* 27 s
- 5) Dari hasil tuning PID hasil yang diinginkan tercapai. Tetapi dengan kebutuhan hidroponik hanya dibutuhkan nilai K_p dan K_i

5.2. Saran

Saran untuk peneliti- peneliti berikutnya yang memiliki topik sejenis dari penelitian ini adalah sebagai berikut:

- 1) Pengamatan pengaruh penggunaan sistem kontrol dilakukan mulai benih hingga panen.
- 2) Ditambahkan pengaruh tanaman terhadap respon kontrol PID pada pH air hidroponik
- 3) Digabungkan dengan disistem kontrol yang lain seperti EC ataupun DO

Halaman ini memang dikosongkan

DAFTAR PUSTAKA

- [1] Arduino. 2013. Arduino Mega 2560. <http://arduino.cc/en/Main/arduinoBoardMega2560> diakses tanggal 2 Mei 2018
- [2] Axelson, J., (1997), The Microcontroller Idea Book Circuits, Programs, & Applications featuring the 8052-BASIC Microcontroller, Lakeview Research, Madison, pp. 1-10
- [3] Electrical Engineering.2018.
<https://electronics.stackexchange.com/questions/369108/how-to-tune-pid-controller-for-servo-steering> diakses tanggal 1 Juni 2018
- [4].Kustanti, “Pengendalian Kadar Keasaman (pH) Pada Sistem Hidroponik Stroberi Menggunakan Kontroler PID Berbasis Arduino Uno,” pp. 1–6, 2014.
- [5] Larutan Asam dan Basa. Internet :
http://kimia.upi.edu/utama/bahanajar/kuliah_web/2008/RAHAYU_060127/pH.html
- [6] Lingga, P., “Hidroponik Bercocok Tanam Tanpa Tanah. Penebar Swadaya”. Jakarta,2009.
- [7] Marlin, Thomas E. “Process Control, Designing Processes and Control Systems for Dynamic Performance”. McGraw-Hill International Editions. 1999.
- [8]Ogata, K., 1996, Teknik Kontrol Automatik, Jilid 1, Edisi 2, Erlangga, Jakarta.
- [9] Ogata, K., 2010, *Modern Control Engineering Fifth Edition*, Fifth, Prentice Hall, New Jersey..
- [10] Resh, H. M, Hydroponic Food Production, Santa Barbara: Woodbridge Press Pbl., 1998.
- [11] Welander, Peter, “Understanding Derivative in PID Control”, Control Engineering, 2, 24-27, 2010

LAMPIRAN A
FOTO PERANCANGAN HIDROPONIK DAN ALAT
KONTROL



Gambar C.1 Perancangan Sistem Hidroponik



Gambar C.2 pH Buffer



Gambar C.3 Alat Kontrol pH

LAMPIRAN B

PROGRAM PID PADA ARDUINO

```
#ifndef PID_v1_h
#define PID_v1_h
#define LIBRARY_VERSION 1.2.1

class PID
{
    #define AUTOMATIC 1
    #define MANUAL 0
    #define DIRECT 0
    #define REVERSE 1
    #define P_ON_M 0
    #define P_ON_E 1
    #define D_ON_M 0
    #define D_ON_E 1
    PID(double*, double*, double*, double, double, double, int, int,
        int);

    PID(double*, double*, double*,
        double, double, double, int);

    void SetMode(int Mode);

    bool Compute();

    void SetOutputLimits(double, double);

    void SetTunings(double, double,
        double);
```


double, int, int);

void SetControllerDirection(int);

void SetSampleTime(int);

double GetKp();

double GetKi();

double GetKd();

int GetMode();

int GetDirection();

double GetPEout();

double GetPIout();

double GetIout();

double GetDEout();

double GetDIout();

private:

void Initialize();

double dispKp;

double dispKi;

double dispKd;

double dispPEout;

double dispPIout;

double dispIout;

double dispDEout;

double dispDIout;

double kp;

double ki;

```

double kd;

    int controllerDirection;
    int pOn;
    int dOn;

double *myInput;
double *myOutput;
double *mySetpoint;


    unsigned long lastTime;
    double outputSum, lastInput, lastError;

    unsigned long SampleTime;
    double outMin, outMax;
    bool inAuto, pOnE, dOnE;
};
#endif


#if ARDUINO >= 100
    #include "Arduino.h"
#else
    #include "WProgram.h"
#endif
#include <PID_v1.h>
PID::PID(double* Input, double* Output, double* Setpoint,
        double Kp, double Ki, double Kd, int POn, int DOn, int
        ControllerDirection)
{

```

```

myOutput = Output;
myInput = Input;
mySetpoint = Setpoint;
inAuto = false;
PID::SetOutputLimits(0, 255);
SampleTime = 100;

PID::SetControllerDirection(ControllerDirection);
PID::SetTunings(Kp, Ki, Kd, POn, DOn);

lastTime = millis()-SampleTime;
}
PID::PID(double* Input, double* Output, double* Setpoint,
         double Kp, double Ki, double Kd, int ControllerDirection)
:PID::PID(Input, Output, Setpoint, Kp, Ki, Kd, P_ON_E,
         D_ON_E, ControllerDirection)
{

}

bool PID::Compute()
{
    if(!inAuto) return false;
    unsigned long now = millis();
    unsigned long timeChange = (now - lastTime);
    if(timeChange>=SampleTime)
    {
        double input = *myInput;
        double error = *mySetpoint - input;
        double dInput = (input - lastInput);
        double dError = (error - lastError);
        outputSum+= (ki * error);
    }
}

```

```

        if(!pOnE) outputSum -= kp * dInput;

        if(outputSum > outMax) outputSum = outMax;
        else if(outputSum < outMin) outputSum = outMin;

        /*Add Proportional on Error, if P_ON_E is specified*/
        double output;
        if(pOnE) output = kp * error;
        else output = 0;
        if(dOnE) output += outputSum + kd * dError;
        else output += outputSum - kd * dInput;

        if(output > outMax) output = outMax;
        else if(output < outMin) output = outMin;
        *myOutput = output;
        lastInput = input;
        lastError = error;
        lastTime = now;

        dispPEout = kp * error;
        dispPIout = -kp * dInput;
        dispIout = outputSum;
        dispDEout = kd * dError;
        dispDIout = -kd * dInput;

        return true;
    }
    else return false;
}

```

```

void PID::SetTunings(double Kp, double Ki, double Kd, int POn,
    int DOn)
{
    if (Kp<0 || Ki<0 || Kd<0) return;

    pOn = POn;
    dOn = DOn;
    pOnE = POn == P_ON_E;
    dOnE = DOn == D_ON_E;

    dispKp = Kp; dispKi = Ki; dispKd = Kd;

    double SampleTimeInSec = ((double)SampleTime)/1000;
    kp = Kp;
    ki = Ki * SampleTimeInSec;
    kd = Kd / SampleTimeInSec;

    if(controllerDirection ==REVERSE)
    {
        kp = (0 - kp);
        ki = (0 - ki);
        kd = (0 - kd);
    }
}

void PID::SetTunings(double Kp, double Ki, double Kd){
    SetTunings(Kp, Ki, Kd, pOn, dOn);
}

void PID::SetSampleTime(int NewSampleTime)
{
    if (NewSampleTime > 0)
    {

```

```

        double ratio = (double)NewSampleTime
                        / (double)SampleTime;
        ki *= ratio;
        kd /= ratio;
        SampleTime = (unsigned long)NewSampleTime;
    }
}

void PID::SetOutputLimits(double Min, double Max)
{
    if(Min >= Max) return;
    outMin = Min;
    outMax = Max;

    if(inAuto)
    {
        if(*myOutput > outMax) *myOutput = outMax;
        else if(*myOutput < outMin) *myOutput = outMin;

        if(outputSum > outMax) outputSum= outMax;
        else if(outputSum < outMin) outputSum= outMin;
    }
}

void PID::SetMode(int Mode)
{
    bool newAuto = (Mode == AUTOMATIC);
    if(newAuto && !inAuto)
    { /*we just went from manual to auto*/
        PID::Initialize();
    }
}

```

```

    inAuto = newAuto;
}
void PID::Initialize()
{
    outputSum = *myOutput;
    lastInput = *myInput;
    if(outputSum > outMax) outputSum = outMax;
    else if(outputSum < outMin) outputSum = outMin;
}
void PID::SetControllerDirection(int Direction)
{
    if(inAuto && Direction !=controllerDirection)
    {
        kp = (0 - kp);
        ki = (0 - ki);
        kd = (0 - kd);
    }
    controllerDirection = Direction;
}

```

```

double PID::GetKp(){ return  dispKp; }
double PID::GetKi(){ return  dispKi; }
double PID::GetKd(){ return  dispKd; }
double PID::GetPEout(){return dispPEout;}
double PID::GetPIout(){return dispPIout;}
double PID::GetIout(){return dispIout;}
double PID::GetDEout(){return dispDEout;}
double PID::GetDIout(){return dispDIout;}
int PID::GetMode(){ return  inAuto ? AUTOMATIC : MANUAL;}
int PID::GetDirection(){ return controllerDirection;}

```

LAMPIRAN C

SOURCE CODE FIRMWARE

```
//Libraries
#include <Wire.h>
#include <Adafruit_ADS1015.h>
#include <PID_v1.h>
#include <LiquidCrystal_I2C.h>
#include "SdFat.h"
#include <SPI.h>
#include "RTCLib.h"
#include <Keypad.h>

//Pin addressing
//Hydroponic 1
#define H1Up    8
#define H1Down  9
#define H1EC    10
#define H1Lv1   54
#define H1Sol   31
#define H1Led   23
#define H1Pump  62
#define H1Mix   63

#define SdCS    39
#define WifiCS  41
#define LanCS   43
#define max6675CS 47
#define fanPwm  45

//ADC Setup
Adafruit_ADS1115 ads1(0x48);
```



```
int16_t adc0, adc1, adc2, adc3;
```

```
//Measurement Setup
```

```
float pH1, EC1;
```

```
unsigned long timeLastMeasure;
```

```
int timeIntervalMeasure = 200;
```

```
//PID Control Setup (D_ON_E = 1, D_ON_M = 0)
```

```
double In_pH1, Out_pH1, Sp_pH1, Kp_pH1 = 10, Ki_pH1 = 0,  
      Kd_pH1 = 0;
```

```
double In_EC1, Out_EC1, Sp_EC1, Kp_EC1 = 10, Ki_EC1 = 0,  
      Kd_EC1 = 0;
```

```
PID PID_pH1(&In_pH1, &Out_pH1, &Sp_pH1, Kp_pH1,  
            Ki_pH1, Kd_pH1, P_ON_E, D_ON_E, DIRECT);
```

```
PID PID_EC1(&In_EC1, &Out_EC1, &Sp_EC1, Kp_EC1,  
            Ki_EC1, Kd_EC1, P_ON_E, D_ON_E, DIRECT);
```

```
float percent_pH1, percent_EC1;
```

```
int Pstate_pH1 = 1, Dstate_pH1 = 1;
```

```
//Watering & Led Control Setup
```

```
bool waterTime1 = true;
```

```
bool waterModel1;
```

```
int lightTime1[] = {17, 30, 5, 45};
```

```
String newLight = "00.00-00.00";
```

```
//LCD Setup
```

```
LiquidCrystal_I2C lcd(0x3F, 20, 4);
```

```
unsigned long timeLastLCD;
```

```
int timeIntervalLCD = 1000;
```

```

//SD Setup
SdFat SD;
File myFile;
unsigned long timeLastSD;
int timeIntervalSD = 1000;
char fileName[] = "00-00-00.CSV";
int SDstate = 0, SDmode = 1;
int lastHour, lastMinute;

//RTC Setup
RTC_DS3231 rtc;
DateTime now;
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday",
    "Wednesday", "Thursday", "Friday", "Saturday"};

//Keypad Setup
const byte ROWS = 4, COLS = 3;
char numKey[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}};
byte rowPins[ROWS] = {34, 32, 30, 28};
byte colPins[COLS] = {26, 24, 22};
Keypad numPad = Keypad(makeKeymap(numKey), rowPins,
    colPins, ROWS, COLS);
char key;

//Menu Setup
char mode = '0';
char newParameter[] = "00.00";

```

```
bool parameterChange = 0;
int page = 0, setPage = 0;
bool Setup = true, ctrlSetup = true;
bool m = 0;
int k = 0;
```

```
const int numReading = 5;
int pH1read[numReading];
int j = 0;
int i = 0;
double total = 0;
double pH1_V;
const int data = 5;
float pH[data];
```

```
void setup() {
  Serial.begin(9600);
  for (int a = 2; a <= 69; a++){
    if (a == 14) {a = 23;}
    else if (a == 47) {a = 58;}
    if (a > 23 && a < 35) {a++;}
    pinMode(a, OUTPUT);
    if (a >= 23 && a <= 69) {
      if ((!(a >= 39 && a <= 46)) || (a == 40 || a == 42))
        {digitalWrite(a, HIGH);}
    }
  }
  pinMode(53, OUTPUT);
```

```
  rtc.begin();
  SD.begin(SdCS);
  ads1.begin();
```

```
ads1.setGain(GAIN_ONE);  
lcd.begin(); lcd.backlight();
```

```
lcd.setCursor(5, 1); lcd.print("Hydroponic");  
lcd.setCursor(3, 2); lcd.print("Control System"); delay(2500);  
  lcd.clear();  
displayTime(); delay(3000); lcd.clear();  
lcd.setCursor(4,1); lcd.print("Initializing");
```

```
lcd.setCursor(8,2);  
for (j = 0 ; j < data; j++){  
  for (i; i < numReading; i++){  
    pH1read[i] = ads1.readADC_SingleEnded(0);  
    total += pH1read[i];  
    delay(timeIntervalMeasure);}  
pH1_V = total/numReading*0.000125;  
if (pH1_V <= 1.8542){  
  pH1 = 6.86 - ((1.8369 - pH1_V) / 0.109929825);  
} else {  
  pH1 = 9.18 - ((2.1011 - pH1_V) / 0.11387931);}  
total = 0;  
i = 0;  
  lcd.print(".");}  
j = 0;  
lcd.clear(); lcd.setCursor(7,1); lcd.print("READY!"); delay(1000);  
  lcd.clear();
```

```
Sp_pH1 = 6.5; Sp_EC1 = 2000;  
PID_pH1.SetSampleTime(1000);  
PID_EC1.SetSampleTime(1000);  
PID_pH1.SetOutputLimits(-225, 225);
```

```

PID_EC1.SetOutputLimits(0, 225);

numPad.begin(makeKeymap(numKey));
numPad.addEventListener(numPadEvent);
numPad.setHoldTime(1800);
lcd.clear();
}

void loop() {
  if (Setup == true){initialSetup();}
  key = numPad.getKey();

  measurement();
  control();
  displayLCD();
  dataWrite();

  if (j >= data){j = 0;}
}

void initialSetup(){
  if (!SD.begin(SdCS)){
    lcd.setCursor(6, 1); lcd.print("Storage");
    lcd.setCursor(3, 2); lcd.print("not available!");
    delay(2000);
  } else {
    lcd.clear(); getFileName();
    lcd.setCursor(2, 0); lcd.print("Storage detected"); delay(1000);
    lcd.setCursor(2, 1); lcd.print("Save data to SD?");
    lcd.setCursor(5, 3); lcd.print("*No  #Yes");
    while (SDstate == 0){
      char SDKey = numPad.getKey();

```

```

    if (SDKey == '*'){
        SDstate = 1;
    } else if (SDKey == '#'){
        SDstate = 2;}
}
if (SDstate == 1){
    lcd.clear(); lcd.setCursor(3, 1); lcd.print("Data not saved");
} else if (SDstate == 2){
    lcd.clear(); lcd.setCursor(3, 1); lcd.print("Data saved to:");
    lcd.setCursor(4, 2); lcd.print(fileName);
}
myFile = SD.open(fileName, FILE_WRITE);
myFile.println(" ");
myFile.println(".,Hour,Minute,Second,pH PV,pH
    MV,Pout,Iout,Dout,.,EC PV,EC MV,Pout,Iout,Dout,");
myFile.close();
}
delay(1500); lcd.clear();
while (ctrlSetup == true){
    lcd.setCursor(2, 0); lcd.print(" Activate Control");
    lcd.setCursor(6, 1); lcd.print("System?");
    lcd.setCursor(5, 3); lcd.print("*OFF  #ON");
    char ctrlKey = numPad.getKey();
    if (ctrlKey == '*'){
        PID_pH1.SetMode(MANUAL);
        lcd.clear(); lcd.setCursor(5, 1); lcd.print("MONITORING");
        lcd.setCursor(8, 2); lcd.print("ONLY");
        ctrlSetup = false;
    } else if (ctrlKey == '#'){
        PID_pH1.SetMode(AUTOMATIC);
        printPID();
    }
}

```

```

    lcd.clear(); lcd.setCursor(7, 1); lcd.print("CONTROL");
    lcd.setCursor(6, 2); lcd.print("ACTIVATED");
    ctrlSetup = false;
  }
}
delay(1000); lcd.clear(); Setup = false;
}

void measurement(){
  unsigned long timeNowMeasure = millis();
  int timeChangeMeasure = (timeNowMeasure - timeLastMeasure);
  if (timeChangeMeasure >= timeIntervalMeasure){
    pH1read[i] = ads1.readADC_SingleEnded(0);
    total += pH1read[i];
    i++;
    if (i >= numReading){
      pH1_V = total/numReading*0.000125;

      if (pH1_V <= 1.8542){
        pH1 = 6.86 - ((1.8369 - pH1_V) / 0.109929825);
      } else {
        pH1 = 9.18 - ((2.1011 - pH1_V) / 0.11387931);
      }

      total = 0;
      i = 0;
      j++;}

    Serial.println(pH1_V, 4);
    timeLastMeasure = timeNowMeasure;
  }
}

void control(){

```

```

//Hydroponic 1
{ In_pH1 = pH1;
  PID_pH1.Compute();
  percent_pH1 = map(Out_pH1, 0, 225, 0, 1000);
  if (percent_pH1/10 > 1){
    analogWrite(H1Up, Out_pH1+30);
    analogWrite(H1Down, 0);
  } else if (percent_pH1/10 < -1){
    analogWrite(H1Down, abs(Out_pH1-30));
    analogWrite(H1Up, 0);
  } else {
    analogWrite(H1Up, 0);
    analogWrite(H1Down, 0);}

  if (waterTime1 == true){
    digitalWrite(H1Pump, LOW);
  } else {
    digitalWrite(H1Pump, HIGH);}

  if (now.hour() == lightTime1[0]){
    if (now.minute() >= lightTime1[1]){
      digitalWrite(H1Led, LOW);
    } else {
      digitalWrite(H1Led, HIGH);
    }
  } else if (now.hour() == lightTime1[2]){
    if (now.minute() <= lightTime1[3]){
      digitalWrite(H1Led, LOW);
    } else {
      digitalWrite(H1Led, HIGH);
    }
  }
}

```



```

    } else if (now.hour() > lightTime1[0] || now.hour() <
        lightTime1[2]){
        digitalWrite(H1Led, LOW);
    } else {
        digitalWrite(H1Led, HIGH);
    }
}
}
}

```

```

void displayLCD(){
    unsigned long timeNowLCD = millis();
    int timeChangeLCD = (timeNowLCD - timeLastLCD);
    if (page == 1){timeIntervalLCD = 333;} else {timeIntervalLCD =
        1000;}
    if (timeChangeLCD >= timeIntervalLCD){
        if (page == 0){
            lcd.setCursor(0, 0);
            if (PID_pH1.GetMode() == MANUAL){
                lcd.print("[OFF]");
            } else {
                lcd.print("[PID]");
            }
            lcd.setCursor(0, 1); lcd.print("Kp "); lcd.print(Kp_pH1, 1);
            lcd.setCursor(0, 2); lcd.print("Ki "); lcd.print(Ki_pH1, 2);
            lcd.setCursor(0, 3); lcd.print("Kd "); lcd.print(Kd_pH1, 2);
            lcd.setCursor(8, 0); lcd.print("SP "); lcd.print(Sp_pH1);
            lcd.setCursor(8, 1); lcd.print("PV "); lcd.print(pH1, 2);
            lcd.setCursor(8, 2); lcd.print("Er"); if ((Sp_pH1-In_pH1)>=0)
                {lcd.print(" ");} if (Sp_pH1-In_pH1>=10) {lcd.print(Sp_pH1-
                In_pH1, 1);} else {lcd.print(Sp_pH1-In_pH1, 2);}
            lcd.setCursor(8, 3); lcd.print("MV");

            if (percent_pH1/10>=0){lcd.print(" ");}

```

```
if (percent_pH1/10>=10){lcd.print(percent_pH1/10, 1);}
else {lcd.print(percent_pH1/10, 2);} lcd.print("%");
```

```
} else if (page == 1){
if (setPage != 0 && parameterChange ==
false){lcd.setCursor(1, 3); lcd.print("*bck      #chg");}
if (parameterChange == true){
if ((setPage >1 && setPage <6) || setPage == 8){
lcd.setCursor(0, 3); lcd.print("*N    <<* #>>  #Y");
} else if (setPage == 1||setPage == 7){
lcd.setCursor(0, 3); lcd.print("*N          #Y");
}
}
}
```

```
if (setPage == 0){
lcd.setCursor(5, 0); lcd.print("[SETTINGS]");
lcd.setCursor(0, 1); lcd.print("1:Mode");
lcd.setCursor(0, 2); lcd.print("2:SP");
lcd.setCursor(7, 1); lcd.print("3:Kp");
lcd.setCursor(7, 2); lcd.print("4:Ki");
lcd.setCursor(7, 3); lcd.print("5:Kd");
lcd.setCursor(13, 1); lcd.print("6:SD");
lcd.setCursor(13, 2); lcd.print("7:Water");
lcd.setCursor(13, 3); lcd.print("8:Light");
```

```
} else if (setPage == 1){
if (parameterChange == false){
lcd.setCursor(6, 0); lcd.print("[1.MODE]");
if (PID_pH1.GetMode() == MANUAL) {
lcd.setCursor(5, 1); lcd.print("Monitoring");
} else {
```

```

        lcd.setCursor(5, 1); lcd.print("PID Control");}
    } else {
        lcd.setCursor(3, 1); lcd.print("Enable control?");
        lcd.setCursor(4, 2); lcd.print("0:No  1:Yes");
        changeParameter();
    }
} else if (setPage == 2){
    if (parameterChange == false){
        lcd.setCursor(4, 0); lcd.print("[2.SETPOINT]");
        lcd.setCursor(6, 1); lcd.print("SP: "); lcd.print(Sp_pH1, 2);
    } else {
        lcd.setCursor(4, 1); lcd.print("Input new Sp");
        changeParameter();
    }
} else if (setPage == 3){
    if (parameterChange == false){
        lcd.setCursor(2, 0); lcd.print("[3.PROPORTIONAL]");
        lcd.setCursor(6, 1); lcd.print("Kp: "); lcd.print(Kp_pH1, 1);
    } else {
        lcd.setCursor(4, 1); lcd.print("Input new Kp");
        changeParameter();
    }
} else if (setPage == 4){
    if (parameterChange == false){
        lcd.setCursor(4, 0); lcd.print("[4.INTEGRAL]");
        lcd.setCursor(6, 1); lcd.print("Ki: "); lcd.print(Ki_pH1, 2);
    } else {
        lcd.setCursor(4, 1); lcd.print("Input new Ki");
        changeParameter();
    }
} else if (setPage == 5){
    if (parameterChange == false){

```

```

    lcd.setCursor(3, 0); lcd.print("[5.DERIVATIVE]");
    lcd.setCursor(6, 1); lcd.print("Kd: "); lcd.print(Kd_pH1, 2);
  } else {
    lcd.setCursor(4, 1); lcd.print("Input new Kd");
    changeParameter();
  }
} else if (setPage == 6){
  if (parameterChange == false){
    lcd.setCursor(4, 0); lcd.print("[6.STORAGE]");
    if (!SD.begin(SdCS)){
      lcd.setCursor(3, 1); lcd.print("  Storage  ");
      lcd.setCursor(3, 2); lcd.print("not available!");
    } else if (SDstate == 0||SDstate == 1){
      lcd.setCursor(3, 1); lcd.print("Data not saved");
      lcd.setCursor(3, 2); lcd.print("          ");
    } else if (SDstate == 2){
      lcd.setCursor(3, 1); lcd.print("Data saved in:");
      lcd.setCursor(4, 2); lcd.print(fileName);
    }
  } else {
    lcd.setCursor(2, 1); lcd.print("Save data to SD?");
    lcd.setCursor(3, 2); lcd.print(" 0:No   1:Yes ");
    changeParameter();
  }
} else if (setPage == 7){
  if (parameterChange == false){
    lcd.setCursor(4, 0); lcd.print("[7.WATERING]");
    if (waterTime1 == true){
      lcd.setCursor(5, 1); lcd.print("Pump active");
    } else {
      lcd.setCursor(4, 1); lcd.print("Pump inactive");
    }
  }
}

```

```

    }
} else {
    lcd.setCursor(3, 1); lcd.print("Activate pump?");
    lcd.setCursor(4, 2); lcd.print("0:No 1:Yes");
    changeParameter();
}
} else if (setPage == 8){
    if (parameterChange == false){
        lcd.setCursor(4, 0); lcd.print("[8.LIGHTING]");
        lcd.setCursor(4, 1); if (lightTime1[0]<10){lcd.print('0');}
        lcd.print(lightTime1[0]); lcd.print("."); if
        (lightTime1[1]<10){lcd.print('0');} lcd.print(lightTime1[1]);
        lcd.print(" -"); if (lightTime1[2]<10){lcd.print('0');}
        lcd.print(lightTime1[2]); lcd.print("."); if
        (lightTime1[3]<10){lcd.print('0');} lcd.print(lightTime1[3]);
    } else {
        lcd.setCursor(4, 1); lcd.print("Set new time");
        changeParameter();
    }
}

} else {
    lcd.setCursor(14, 2); lcd.print("QUATIX");
    lcd.setCursor(6, 3); lcd.print("COPYRIGHT 2018");}

timeLastLCD = timeNowLCD;
}
}

void dataWrite(){
    now = rtc.now();
    int nowHour = now.hour(); int nowMinute = now.minute();

```

```

unsigned long timeNowSD = millis();
int timeChangeSD = (timeNowSD - timeLastSD);
if (timeChangeSD >= timeIntervalSD){
  if (SDstate == 2){
    String dataLog;
    if (nowHour != lastHour){
      dataLog = "," + String(now.hour()) + ',';
      lastHour = nowHour;
    } else {
      dataLog = ".,,";
    }

    if (nowMinute != lastMinute){
      dataLog += String(now.minute());
      lastMinute = nowMinute;
    }

    dataLog += ',' + String(now.second()) + "," + String(In_pH1) + ','
      + String(Out_pH1) + ',';

    if (Pstate_pH1 == 1){
      dataLog += String(PID_pH1.GetPEout()) + ',';
    } else {
      dataLog += String(PID_pH1.GetPIout()) + ',';
    }

    dataLog += String(PID_pH1.GetIout()) + ',';

    if (Dstate_pH1 == 1){
      dataLog += String(PID_pH1.GetDEout());
    } else {
      dataLog += String(PID_pH1.GetDIout());
    }

    myFile = SD.open(fileName, FILE_WRITE);
  }
}

```

```
myFile.println(dataLog);  
myFile.close();}
```

```
timeLastSD = timeNowSD;}  
}
```

```
void printPID(){  
  myFile = SD.open(fileName, FILE_WRITE);  
  myFile.print("Sp =,"); myFile.println(Sp_pH1);  
  myFile.print("Kp =,"); myFile.println(Kp_pH1);  
  myFile.print("Ki =,"); myFile.println(Ki_pH1);  
  myFile.print("Kd =,"); myFile.println(Kd_pH1);  
  myFile.close();  
}
```

```
void displayTime(){  
  lcd.setCursor(3, 0); lcd.print("Hello Bajindul");  
  delay(750);  
  now = rtc.now();  
  if (now.hour()>3 && now.hour()<11){  
    lcd.setCursor(4, 1); lcd.print("Good Morning");  
  } else if (now.hour()>10 && now.hour()<16){  
    lcd.setCursor(3, 1); lcd.print("Good Afternoon");  
  } else if (now.hour()>15 && now.hour()<19){  
    lcd.setCursor(4, 1); lcd.print("Good Evening");  
  } else {  
    lcd.setCursor(5, 1); lcd.print("Good Night");}  
  delay(500); lcd.setCursor(9, 2); lcd.print(""); delay(750);
```

```
if  
  (now.dayOfTheWeek()==0||now.dayOfTheWeek()==1||now.d  
   ayOfTheWeek()==5){
```

```

        lcd.setCursor(4, 3);                                //SUNDAY,
        MONDAY, FRIDAY,
    } else if (now.dayOfTheWeek()==2){
        lcd.setCursor(4, 3);                                //TUESDAY
    } else if (now.dayOfTheWeek()==4||now.dayOfTheWeek()==6){
        lcd.setCursor(3, 3);
        //THURSDAY, SATURDAY
    } else {
        lcd.setCursor(3, 3);}
        //WEDNESDAY

    lcd.print(daysOfTheWeek[now.dayOfTheWeek()]);
    lcd.print(" "); if (now.hour()<10) {lcd.print("0");}
        lcd.print(now.hour()); lcd.print(":"); if (now.minute()<10)
        {lcd.print("0");} lcd.print(now.minute());
    }

void getFileName(){
    fileName[6] = (now.year()/10)%10 + '0';
    fileName[7] = now.year()%10 + '0';
    fileName[3] = now.month()/10 + '0';
    fileName[4] = now.month()%10 + '0';
    fileName[0] = now.day()/10 + '0';
    fileName[1] = now.day()%10 + '0';
}

void numPadEvent(KeypadEvent key){
    switch (numPad.getState()){

    case PRESSED:
        if (key == '*'){
            if (page == 1 && parameterChange == false){

```



```

    setPage = 0;
    lcd.clear();
}
if (parameterChange == true){
    if (setPage >1 && setPage <6){
        k--;
        if (k < 0){k = 4;}
        else if (k > 4){k = 0;}
        else if (k == 2){k = 1;}
    } else if (setPage == 8){
        k--;
        if (k < 0){k = 10;}
        else if (k > 10){k = 0;}
        else if (k == 2){k = 1;}
        else if (k == 5){k = 4;}
        else if (k == 8){k = 7;}
    }
}

} else if (key == '#'){
    if (parameterChange == true){
        if (setPage >1 && setPage <6){
            k++;
            if (k < 0){k = 4;}
            else if (k > 4){k = 0;}
            else if (k == 2){k = 3;}
        } else if (setPage == 8){
            k++;
            if (k < 0){k = 10;}
            else if (k > 10){k = 0;}
            else if (k == 2){k = 3;}
            else if (k == 5){k = 6;}
        }
    }
}

```

```

        else if (k == 8){k = 9;}
    }
}
if (page == 1 && setPage != 0){
    parameterChange = true;
}

} else if (key == '0'){
    if (parameterChange == true){
        if (setPage == 1){
            mode = int(key);
        } else if (setPage == 6){
            SDmode = 1;
        } else if (setPage == 7){
            waterModel = false;
        } else if (setPage == 8){
            newLight[k] = int(key);
        } else {
            newParameter[k] = int(key);
        }
    }
}

} else if (key == '1'){
    if (page == 1 && parameterChange == false){
        setPage = 1;
        lcd.clear();
    }
    if (parameterChange == true){
        if (setPage == 1){
            mode = int(key);
        } else if (setPage == 6){

```

```

        SDmode = 2;
    } else if (setPage == 7){
        waterModel = true;
    } else if (setPage == 8){
        newLight[k] = int(key);
    } else {
        newParameter[k] = int(key);
    }
}

} else if (key == '2'){
    if (page == 1 && parameterChange == false){
        setPage = 2;
        lcd.clear();
    }
    if (parameterChange == true){
        if (setPage == 8){
            newLight[k] = int(key);
        } else {
            newParameter[k] = int(key);
        }
    }
}

} else if (key == '3'){
    if (page == 1 && parameterChange == false){
        setPage = 3;
        lcd.clear();
    }
    if (parameterChange == true){
        if (setPage == 8){
            newLight[k] = int(key);
        } else {

```

```
        newParameter[k] = int(key);
    }
}

} else if (key == '4'){
    if (page == 1 && parameterChange == false){
        setPage = 4;
        lcd.clear();
    }
    if (parameterChange == true){
        if (setPage == 8){
            newLight[k] = int(key);
        } else {
            newParameter[k] = int(key);
        }
    }
}

} else if (key == '5'){
    if (page == 1 && parameterChange == false){
        setPage = 5;
        lcd.clear();
    }
    if (parameterChange == true){
        if (setPage == 8){
            newLight[k] = int(key);
        } else {
            newParameter[k] = int(key);
        }
    }
}

} else if (key == '6'){
```

```

if (page == 1 && parameterChange == false){
    setPage = 6;
    lcd.clear();
}
if (parameterChange == true){
    if (setPage == 8){
        newLight[k] = int(key);
    } else {
        newParameter[k] = int(key);
    }
}

} else if (key == '7'){
    if (page == 1 && parameterChange == false){
        setPage = 7;
        lcd.clear();
    }
    if (parameterChange == true){
        if (setPage == 8){
            newLight[k] = int(key);
        } else {
            newParameter[k] = int(key);
        }
    }
}

} else if (key == '8'){
    if (page == 1 && parameterChange == false){
        setPage = 8;
        lcd.clear();
    }
    if (parameterChange == true){
        if (setPage == 8){

```



```

        newParameter[u] = '0';}
    } else if (setPage == 8){
        for (int u = 0; u <=10; u++){
            if (u == 2||u == 5||u == 8){
                u++;}
            newLight[u] = '0';}
        }
    parameterChange = false; k = 0; lcd.clear();
}
}

if (key == '#'){
    if (setPage == 0){
        page += 1;
        lcd.clear();
        if (page == 3){page = 0;}
    }
    if (parameterChange == true){
        if (setPage == 1){
            myFile = SD.open(fileName, FILE_WRITE);
            if (mode == 49){
                PID_pH1.SetMode(AUTOMATIC);
                myFile.println("PID:,ON");
            } else if (mode == 48){
                PID_pH1.SetMode(MANUAL);
                myFile.println("PID:,OFF");
            }
            myFile.close();
            Out_pH1 = 0;
        } else if (setPage >1 && setPage < 6){
            float l = atof(newParameter);
            if (setPage == 2){

```

```

    Sp_pH1 = 1;
} else if (setPage == 3){
    Kp_pH1 = 1;
} else if (setPage == 4){
    Ki_pH1 = 1;
} else if (setPage == 5){
    Kd_pH1 = 1;
}
PID_pH1.SetTunings(Kp_pH1, Ki_pH1, Kd_pH1);
printPID();
for (int u = 0; u <=4; u++){
    if (u == 2){
        u++;}
    newParameter[u] = '0';}
} else if (setPage == 6){
    lcd.clear();
    if (!SD.begin(SdCS)){
        lcd.setCursor(6, 1); lcd.print("Storage");
        lcd.setCursor(3, 2); lcd.print("not available!");
        delay(1000); lcd.clear();
        lcd.setCursor(4, 1); lcd.print("Please insert");
        lcd.setCursor(6, 2); lcd.print("SD Card!");
    } else {
        SDstate = SDmode;
        getFileName();
        myFile = SD.open(fileName, FILE_WRITE);
        if (SDmode == 2){
            myFile.println("SD:,ON");
            lcd.setCursor(3, 1); lcd.print("Data saved to:");
            lcd.setCursor(4, 2); lcd.print(fileName);
        } else {

```



```

        myFile.println("SD:,OFF");
        lcd.setCursor(3, 1); lcd.print("Data not saved");
        lcd.setCursor(5, 2); lcd.print("in SD card!");}
    myFile.close();
}

} else if (setPage == 7){
    waterTime1 = waterModel1;
} else if (setPage == 8){
    char newLightArray[12];
    char *hourBegin;
    char *minuteBegin;
    char *hourEnd;
    char *minuteEnd;
    newLight.toCharArray(newLightArray, 12);
    hourBegin = strtok(newLightArray, ".");
    minuteBegin = strtok(NULL, "-");
    hourEnd = strtok(NULL, ".");
    minuteEnd = strtok(NULL, "\\0");
    lightTime1[0] = atoi(hourBegin);
    lightTime1[1] = atoi(minuteBegin);
    lightTime1[2] = atoi(hourEnd);
    lightTime1[3] = atoi(minuteEnd);
    for (int u = 0; u <=10; u++){
        if (u == 2||u == 5||u == 8){
            u++;}
        newLight[u] = '0';}
    }

    if (setPage != 6){lcd.clear(); lcd.setCursor(7, 1);
    lcd.print("Change"); lcd.setCursor(6, 2);
    lcd.print("Success!");}

```

```

        delay(1000); lcd.clear();
        parameterChange = false;
        page = 0;
        setPage = 0;
        k = 0;
    }
}

break;
}
}

void changeParameter(){
    if (m == 0){
        if (setPage == 1||setPage == 6||setPage == 7){
            lcd.setCursor(9+k, 3);
        } else if (setPage == 8){
            lcd.setCursor(5+k, 2);
        } else {
            lcd.setCursor(8+k, 2);
        }
        lcd.print("_");
        m = 1;
    } else {
        if (setPage == 1){
            lcd.setCursor(9, 3); lcd.print(mode);
        } else if (setPage == 6){
            lcd.setCursor(9, 3); lcd.print(SDmode-1);
        } else if (setPage == 7){
            lcd.setCursor(9, 3); lcd.print(waterModel1);
        } else if (setPage == 8){

```

```
    lcd.setCursor(5, 2); lcd.print(newLight);  
  } else {  
    lcd.setCursor(8, 2); lcd.print(newParameter);  
  }  
  m = 0;  
}  
}
```

BIODATA PENULIS



Nama lengkap penulis adalah Zendy Zita Anggani, lahir di Bontang tanggal 4 Agustus 1993. Penulis telah menyelesaikan pendidikan formal, yaitu SD 1 Yayasan Pupuk Kalimantan Timur, SMP Yayasan Pupuk Kalimantan Timur, SMA Yayasan Pupuk Kalimantan Timur. Setelah lulus dari SMA, penulis diterima di Departemen Teknik Fisika ITS. Selama menempuh pendidikan di ITS, penulis juga

aktif pada kegiatan di luar perkuliahan, seperti mengikuti organisasi tingkat jurusan dan berbagai kepanitiaan yang diadakan di tingkat jurusan. Penulis memiliki ketertarikan dalam penelitian berkaitan dengan teknologi tepat guna dibidang pertanian, perkebunan, dan perikanan. Lingkup teknologi yang dikuasai penulis meliputi instrumentasi dan kontrol, otomasi, optimasi, dan *artificial intelligence*. Bagi pembaca yang memiliki kritik, saran atau ingin berdiskusi seputar topik tersebut, dapat menghubungi penulis dengan melalui email: zendyzitaanggani@gmail.com atau menghubungi di nomor berikut 081230610585.